<중제목>1. 예제 프로그램

〔예제 13-1〕 server: ex13-01s.c

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <sys/utsname.h>
#include <netdb.h>

#define SIZE    sizeof(struct sockaddr_in)

main()
{
    int sockfd;
    char msg;

    struct utsname info;
    struct hostent *hent;

    struct sockaddr_in server = {AF_INET, 2007, INADDR_ANY};
    struct sockaddr_in client;
    int client_len = SIZE;

    uname(&info);
    printf("node name : %s\n", info.nodename);

    hent = gethostbyname(info.nodename);
    printf("official name : %s\n", hent->h_name);

    sockfd = socket(AF_INET, SOCK_DGRAM, 0);

    bind(sockfd, (struct sockaddr *)&server, SIZE);

    recvfrom(sockfd, &msg, 1, 0, (struct sockaddr *)&client, &client_len);

    printf("recv from client : %c\n", msg);

    sendto(sockfd, &msg, 1, 0, (struct sockaddr *)&client, client_len);
```

```
    close(sockfd);
}
```

[예제 13-1-1] client: ex13-01c.c

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <unistd.h>

#define SIZE    sizeof(struct sockaddr_in)

main()
{
    int sockfd;
    char msg, hostname[1024], *ipaddr;

    struct hostent *hent;
    struct sockaddr_in client = {AF_INET, INADDR_ANY, INADDR_ANY};
    struct sockaddr_in server = {AF_INET, 2007};
    int server_len = SIZE;

    gethostname(hostname, 1024);
    printf("hostname : %s\n", hostname);

    hent = gethostbyname(hostname);
    ipaddr = inet_ntoa(*((struct in_addr *)*(hent->h_addr_list)));
    printf("official name : %s\n", hent->h_name);
    printf("IP address : %s\n", ipaddr);

    server.sin_addr.s_addr = inet_addr(ipaddr);

    sockfd = socket(AF_INET, SOCK_DGRAM, 0);

    msg = 'A';
    sendto(sockfd, &msg, 1, 0, (struct sockaddr *)&server, server_len);
    recvfrom(sockfd, &recv, 1, 0, (struct sockaddr *)&server, &server_len);
    printf("recv from server : %c\n", msg);
    close(sockfd);
}
```

[예제 13-2] server: ex13-02.c

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define SIZE    sizeof(struct sockaddr_in)
#define MSGSIZE 1024

main()
{
    int sockfd;
    char msg[MSGSIZE];

    struct sockaddr_in server = {AF_INET, 2007, INADDR_ANY};

    struct sockaddr_in client;
    int client_len = SIZE;

    sockfd = socket(AF_INET, SOCK_DGRAM, 0);

    bind(sockfd, (struct sockaddr *)&server, SIZE);

     recvfrom(sockfd,  &msg,  MSGSIZE,  0,  (struct  sockaddr  *)&client,
&client_len);

    printf("recv from client : %s\n", msg);

     sendto(sockfd, &msg, MSGSIZE, 0, (struct sockaddr *)&client, client_len);

    close(sockfd);
}
```

[예제 13-3] client: ex13-03.c

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define SIZE    sizeof(struct sockaddr_in)
#define MSGSIZE 1024

main()
{
    int sockfd;
    char msg[MSGSIZE], recv[MSGSIZE];

    struct sockaddr_in client = {AF_INET, INADDR_ANY, INADDR_ANY};

    int server_len = SIZE;
    struct sockaddr_in server = {AF_INET, 2007};
    server.sin_addr.s_addr = inet_addr("202.31.200.87");

    sockfd = socket(AF_INET, SOCK_DGRAM, 0);

    strcpy(msg, "Hello world!");

     sendto(sockfd, &msg, MSGSIZE, 0, (struct sockaddr *)&server, server_len);

     recvfrom(sockfd, &recv, MSGSIZE, 0, (struct sockaddr *)&server,
&server_len);

    printf("reply from server: %s\n", recv);

    close(sockfd);
}
```

[예제 13-4] server: ex13-04s.c

```
/* Server */
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define SIZE    sizeof(struct sockaddr_in)

main()
{
    int sockfd;
    char msg, prev;

    struct sockaddr_in server = {AF_INET, 2007, INADDR_ANY};

    struct sockaddr_in client;
    int client_len = SIZE;

    if((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1)
    {
        printf("fail to call socket()\n");
        exit(1);
    }

    if(bind(sockfd, (struct sockaddr *)&server, SIZE) == -1)
    {
        printf("fail to call bind()\n");
        exit(1);
    }

    prev = '\n';
    while(1)
    {
        if(recvfrom(sockfd, &msg, 1, 0,
                    (struct sockaddr *)&client, &client_len) == -1)
        {
            printf("fail to receive message\n");
            continue;
        }
```

```
        printf("%s%c", (prev == '\n') ? "[recv] " : "", msg);
        prev =  msg;

        if(sendto(sockfd, &msg, 1, 0,
                  (struct sockaddr *)&client, client_len) == -1)
       {
            printf("fail to receive message\n");
            continue;
        }
    }
}
```

[예제 13-4-1] client: ex13-04c.c

```
/* Client */
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define SIZE    sizeof(struct sockaddr_in)

main()
{
    int sockfd;
    char msg, prev;

    struct sockaddr_in client = {AF_INET, INADDR_ANY, INADDR_ANY};

    int server_len = SIZE;
    struct sockaddr_in server = {AF_INET, 2007};
    server.sin_addr.s_addr = inet_addr("202.31.200.87");

    if((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1)
    {
        printf("fail to call socket()\n");
        exit(1);
    }

    prev = '\n';
    while(read(0, &msg, 1) != 0)
    {
        if(sendto(sockfd, &msg, 1, 0,
                (struct sockaddr *)&server, server_len) == -1)
        {
            printf("fail to send message\n");
            continue;
        }

        if(recvfrom(sockfd, &msg, 1, 0,
                (struct sockaddr *)&server, &server_len) == -1)
        {
            printf("fail to receive message\n");
```

```
            continue;
        }

        printf("%s%c", (prev == '\n') ? "[recv] " : "", msg);
        prev = msg;
    }
}
```

[예제 13-5] ex13-05.c

```
#include <sys/utsname.h>
#include <unistd.h>

main()
{
        struct utsname info;
        char myname[1024];

        uname(&info);
        printf("sysname: %s\n", info.sysname);
        printf("nodename: %s\n", info.nodename);
        printf("release: %s\n", info.release);
        printf("version: %s\n", info.version);
        printf("machine: %s\n", info.machine);

        gethostname(myname, 1024);
        printf("hostname: %s\n", myname);
}
```

[예제 13-6] ex13-06.c

```
#include <unistd.h>
#include <netdb.h>

main(int argc, char *argv[])
{
    struct hostent *hent;
    char **ptr;

    if(argc < 2) {
        printf("%s hostname\n", argv[0]);
        exit(1);
    }

    if((hent = gethostbyname(argv[1])) == NULL) {
        printf("fail to call gethostbyname()\n");
        exit(1);
    }

    printf("official name : %s\n", hent->h_name);

    for(ptr = hent->h_aliases; *ptr != NULL; *ptr++)
        printf("\talias : %s\n", *ptr);

    if(hent->h_addrtype == AF_INET) {
        ptr = hent->h_addr_list;
        for(; *ptr != NULL; ptr++)
            printf("\taddress : %s\n", inet_ntoa(*((struct in_addr *)*ptr)));
    }
}
```

[예제 13-7] ex13-07.c

```
#include <unistd.h>
#include <netdb.h>

main(int argc, char *argv[])
{
    struct hostent *hent;
    in_addr_t ipaddr;
    char **ptr;

    if(argc < 2) {
        printf("%s ip_address\n", argv[0]);
        exit(1);
    }

    if((ipaddr = inet_addr(argv[1])) == -1) {
        printf("fail to call inet_addr()\n");
        exit(1);
    }

    if((hent = gethostbyaddr((char *)&ipaddr, 4, AF_INET)) == NULL) {
        printf("fail to call gethostbyaddr()\n");
        exit(1);
    }

    printf("official name : %s\n", hent->h_name);

    for(ptr = hent->h_aliases; *ptr != NULL; *ptr++)
        printf("\talias : %s\n", *ptr);

    if(hent->h_addrtype == AF_INET) {
        ptr = hent->h_addr_list;
        for(; *ptr != NULL; ptr++)
            printf("\taddress : %s\n", inet_ntoa(*((struct in_addr *)*ptr)));
    }
}
```